About the Stability of Active Queue Management mechanisms

Giovanni Neglia*, Dario Bauso†, and Laura Giarré†
*Dipartimento di Ing. Elettrica, DIE
Università di Palermo
Palermo, Italia +39-0916615286
Email: {giovanni.neglia@tti.unipa.it,}
†Dip. di Ing. dell'Automazione e dei Sistemi, DIAS
Università di Palermo
Palermo, Italia +39-091481119
Email: {Bauso@ias.unipa.it, giarre@unipa.it}

***Abstract*— In this paper, we highlight that multiple bottlenecks can affect the performance of Active Queue Management (AQM) controllers. These are usually configured on a single bottleneck basis, as if each controller were the only element regulating the TCP traffic along its path. To see this, we consider a network scenario where RED is configured at each router, according to previously developed control theoretic techniques. These configuration rules assure stability in a single bottleneck scenario. Yet, we show that instability may arise when two link become congested. We justify this result through a multiple bottleneck model and give guidelines for new cooperative AQM controllers, exchanging information about the nodes congestion status.**

I. Introduction

AQM has been proposed to support end-to-end TCP congestion control in the Internet [1]. AQM controllers operate at the network nodes to detect incipient congestion and indicate it to TCP sources, which reduce their transmission rate in order to prevent worse congestion. Usually packet drops are used for congestion indication.

Many AQM schemes have been proposed [2], [3], [4], [5]. The proposed algorithms usually rely on some heuristics and their performances appear to be highly dependant on the considered network scenario. Hence, parameter tuning is very difficult (see for example [6], [7], [8], as regards the well-known Random Early Detection -RED- algorithm). Despite the wide literature on AQM, nobody has explicitly taken into account the distributed fashion of TCP flows control across the network: as a matter of fact TCP flows may turn to be controlled at the same time by two or more nodes acting independently according to their AQM settings. According to our opinion, this can hardly affect AQM algorithms performance.

In order to support our thesis we consider RED configuration criteria proposed in [9]. According to the best of our knowledge, these criteria are the only ones based on a theoretical analysis rather than on plausible thumb rules. In [9] the authors start from the single bottleneck nonlinear model for TCP sources proposed in [10]. They linearize it and apply linear feedback control to obtain some RED configuration criteria which guarantee stability under bounded traffic variation (number of flows and round trip time). In this paper, we propose a counterexample to show that RED controllers, configured according to [9], do not prevent from instability if two nodes face congestion at the same time (what we called multi-bottleneck scenario). In other words, we highlight the limits of AQM configuration criteria, that rely only on local information.

This paper is organized as follows. Section II recollects the RED operation and some results from [9], which will be referred to in the following sections. In Section III we present a multi-bottleneck network scenario, that exhibits instability. The presence of instability is derived from performance metrics obtained through simulations.

In Section IV, we provide an analytical insight to better understand the experimental results. In Section V, we discuss issues concerned with i) the robustness of single bottleneck configuration criteria, and ii) the development of new AQM schemes, which take into consideration the distributed nature of AQM control. The last item deals with cooperative controllers, that exchange information about their congestion state. Finally, conclusive remarks and further research issues are given in Section VI.

II. Single bottleneck model

The starting point in [9] is the model described by the following coupled, nonlinear differential equations:

$$\dot{W}(t) = \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))}p(t-R(t)) \qquad (1)$$

$$\dot{q}(t) = \frac{W(t)}{R(t)}N(t) - 1_{q(t)}C \qquad (2)$$

where $1_q = 1$ if $q > 0$, $1_q = 0$ otherwise. Symbols used in the model above are summarized in the following table.

W	expected TCP window size (packets);
q	expected queue length (packets);
R	round-trip time;
C	link capacity (packets/sec);
T_p	propagation delay (secs);
N	load factor (number of TCP sessions);
p	probability of packet drop;

The first equation represents the TCP window, that increase by one every round trip time, and halves when a packet loss occurs. Packet loss rate is computed as the dropping probability times the number of packets sent per time unit. The second equation represents the variation of queue occupancy as the difference between the input traffic and the link capacity.

AQM schemes determine the relation between the dropping probability and the nodes congestion status.

Here we recollect briefly the RED operation. RED gateway calculates the average queue size, x, using a low-pass filter with an exponential weighted moving average of the instantaneous queue, q, as shown in the following recursive formula: $x = \alpha q + (1-\alpha)x$, where α is the memory coefficient. The average queue size is compared with a minimum threshold and a maximum one. When the average queue size is less than the minimum threshold, no packets are dropped. When the average queue size is greater than the maximum threshold, every arriving packet is dropped. When the average queue size is between the minimum and the maximum threshold, each arriving packet is dropped with probability p, where p is a linearly increasing function of the average queue size. RED configuration is then specified through four parameters: the minimum and the maximum threshold (THR_{min}, THR_{max}), the maximum dropping probability in the region of random discard P_{max}, and the aforementioned memory coefficient w_q.

RED can be modelled by the following equations:

$$\dot{x}(t) = -Kx(t) + Kq(t) \qquad (3)$$

$$p(x) = \begin{cases} 0, & 0 \leq x < THR_{min} \\ \frac{x - THR_{min}}{THR_{max} - THR_{min}} P_{max}, & THR_{min} \leq x < THR_{max} \\ 1 & THR_{max} \leq x \end{cases} \qquad (4)$$

TABLE I

Network Parameters

Link	Capacity (Mbps)	Propagation Delay (ms)
1-4	20	15
2-3	10	5
3-4	20	10
4-7	10	10
5-1	20	15
6-2	20	5
8-2	20	15
3-9	20	10

where $K = -\ln(1-\alpha)/\delta$ and δ is the time between two queue samples. The time interval δ can be assumed to be equal to $1/C$ for a congested node.

The linearized system (TCP sources, congested node queue and AQM controller) can be represented by the block diagram of Figure 1. In the block diagram $L = P_{max}/(THR_{max} - THR_{min})$.

The open-loop transfer function of the system in Figure 1 is:

$$F(s) = \frac{L\frac{(RC)^3}{(2N)^2}e^{-sR}}{(1+\frac{s}{K})\left(1+\frac{s}{\frac{2N}{R^2C}}\right)\left(1+\frac{s}{\frac{1}{R}}\right)} \qquad (5)$$

In [9] the authors present RED configuration rules, that guarantee the stability of the linear feedback control system in Figure 1 for $N \geq N^-$ and $R_0 \leq R^+$.

III. An Instability Example

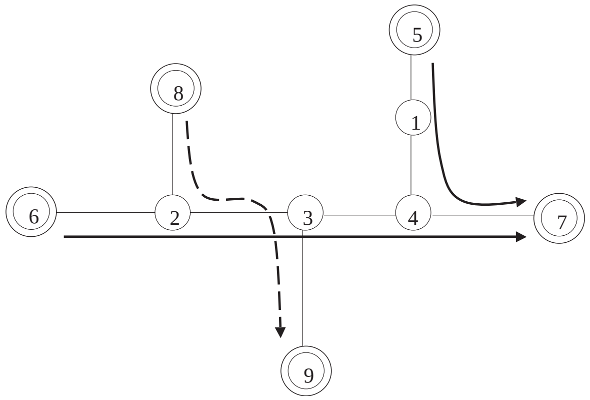

Fig. 2. Network topology

We consider a parking lot network whose topology is depicted in Figure 2. The capacity and the propagation delay of each link are reported in Table I. Packet size is 1500 bytes. Links between nodes 4 and 7 and between nodes 2 and 3 will play the role of bottlenecks.

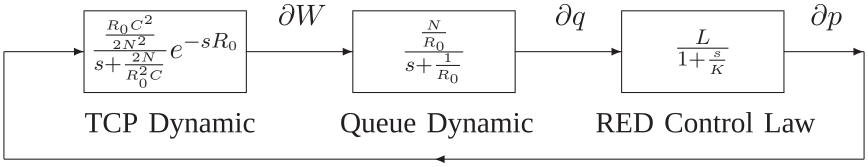
$\frac{\frac{R_0C^2}{2N^2}}{s+\frac{2N}{R_0^2C}}e^{-sR_0}$
∂W
$\frac{\frac{N}{R_0}}{s+\frac{1}{R_0}}$
∂q
$\frac{L}{1+\frac{s}{K}}$
∂p
TCP Dynamic
Queue Dynamic
RED Control Law

Fig. 1. Block diagram of linearized RED control system

The RED algorithm is deployed at nodes 4 and 2, respectively to manage the output queues for the link $4-7$ and $2-3$. In what follows we refer to these buffers simply as node 4 buffer and node 2 buffer, without specifying the link.

Our RED configuration relies on the control theoretic analysis of RED presented in [9]. Nevertheless, we do not adopt exactly the configuration rules proposed there, since their high stability margins do not allow simple counter-example. Then, we verify RED-configuration stability through the Nyquist plot of the open loop transfer function.

We recall that the Nyquist criterion allows one to study the stability of the closed loop system through the polar plot of the open loop transfer function $F(j\omega)$. For the functions we are interested in, the closed loop system is stable if and only if the plot encircles the point $(-1, 0)$.

We choose $THR_{min} = 2$, $THR_{max} = 20$, $P_{max} = 5\%$, and $w_q = 0.002$. This configuration guarantees stability if the number of flows is greater than or equal to $N^- = 7$ and the Round Trip Time is lower than or equal to $110ms$. Figure 3 shows the Nyquist plot of the open loop transfer function (5) for $R = 110ms$ and different number of flows N, whereas Figure 3 shows the Nyquist plot for a number of flows, $N = 8$ and different values of the round trip time, R.

Simulations were conducted through ns v2.27 [12]. We used TCP Reno implementation.

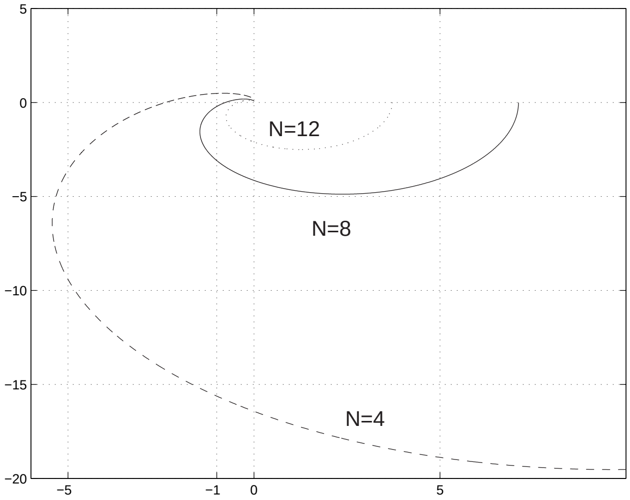

Fig. 3. Nyquist plots for the considered RED configuration and $N = 4, 8, 12$ flows

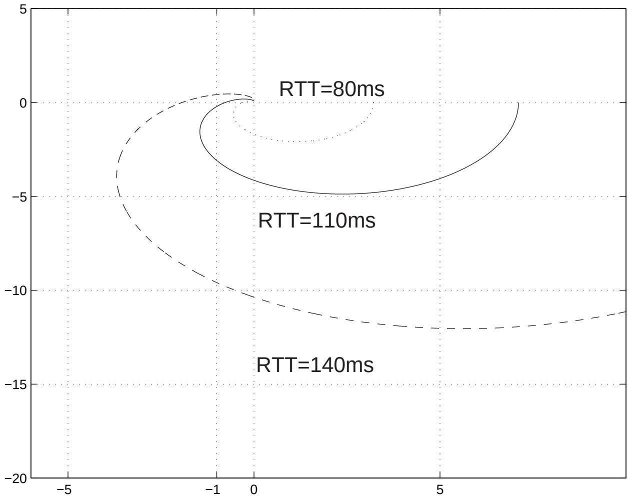

Fig. 4. Nyquist plots for the considered RED configuration and $R = 80, 110, 140$ ms

A. Single Bottleneck

A primary question is which metric is particularly suitable to catch instability phenomena. In this sense, though instability is by many authors addressed looking at the amplitude of queue size oscillations, we will better refer to the normalized standard deviation as a more suitable metric to analyze instability phenomena. For example, when the number of flows decreases, stability margins decrease according to the linear model developed in [9], and one could expect larger queue oscillations. Yet, at the same time the queue average value decreases and the physical constraint of positive queue values can determine smaller oscillations. Ultimately, the cause is the RED coupling of queue length and loss probability, which lets the operating point depend from the network conditions, like the load level. From a control theoretic point of view one says that the RED controller has steady state regulation errors.

Now, in order to analytically show how instability of the linear model concretely affects the network performance, we first present some results regarding the single bottleneck scenario.

Two aggregates, each one of four TCP flows ($N = 8$), enter the network through node 5 and node 6 with destination node 7 (solid lines in figure 2). The link between nodes 4 and 7 is congested.

Figure 5 shows the instantaneous queue occupancy time-plot for the buffer at node 4. RED should be able to keep the queue occupancy within the two thresholds, which are shown in the figure as reference.

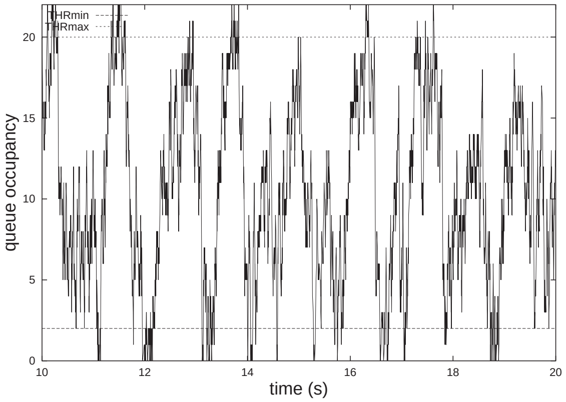

Fig. 5. Instantaneous buffer occupancy with number of flows $N = 8$

Let us progressively reduce the number of flows through the network and see if instability occurs as claimed in [9]. In Figure 6 the buffer occupancy is shown to revisit with an higher frequency the regions associated to buffer overload and underload (out of RED thresholds).

Numerical results for the throughput and the normalized standard deviation are shown in Table II. As the total flow number decrease from 8 to 6 we note that i) the throughput over the link $4-3$ reduces from $9.80 Mbps$ to 9.70, ii) both the average queue occupancy and the oscillation amplitude decrease, respectively from 10.0 to 8.19 and from 5.26 to 4.64, and iii) the normalized standard deviation, i.e. the ratio between standard deviation and mean, increases from 0.52 to 0.56.

If we reduce drastically the number of flows to 4, the above RED configuration, turns to be too aggressive, which is evidenced by higher frequencies of buffer occupancy oscillations and further reduction of the throughput. Even longer periods, where buffer is underloaded results from Figure 7.

Experimental results show that instability predicted by the model in [9] leads to reduced link utilization and higher normalized oscillations (higher jitter in percentage).

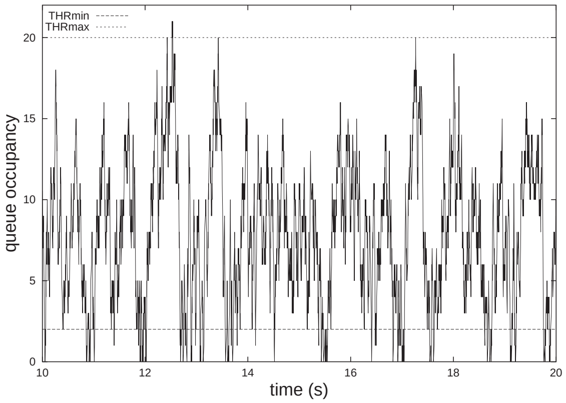

Fig. 6. Instantaneous buffer occupancy with $N = 6$

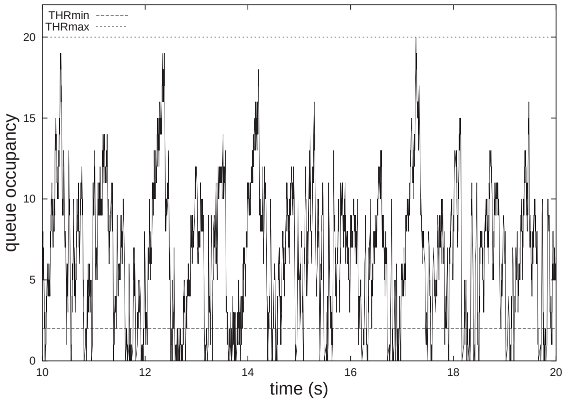

Fig. 7. Instantaneous buffer occupancy with $N = 4$

Conversely, if we increase the number of flows, higher throughput and lower jitter can be achieved.

Node 2 buffer has the same RED configuration. Table II shows similar results when only the link $2-3$ is congested, due to flows coming from nodes 6 and 8.

B. Two Bottlenecks

We now draw the attention to the fact that buffer occupancy instability, may arise when flows through node 4 are in part already controlled by some other congested upstream node, for instance, node 2 when link $2-3$ is congested (see Figure 2).

To recreate artificially such a scenario, let us introduce an additional aggregate entering the network from node 8, with destination node 9 (dotted line in figure 2). Node 4 buffer occupancy for a 4-flows aggregate exhibits a high oscillatory behavior in figure 8.

From Figure 9 instability arises also at node 2.

TABLE II
NUMERICAL RESULTS

N_6	N_5	N_8	Thr_6	Thr_5	Thr_8	$queue_4$ occupancy	$queue_4$ oscillation	$queue_2$ occupancy	$queue_2$ oscillation
6	6	0	5.36	4.57	-	13.6	0.41	0.94	0.26
4	4	0	5.39	4.41	-	10.0	0.52	0.95	0.25
3	3	0	5.29	4.41	-	8.19	0.56	0.96	0.28
2	2	0	5.32	4.17	-	6.31	0.64	0.97	0.43
0	4	0	-	9.49	-	5.51	0.72	0	0
4	0	4	4.92	-	4.92	0	0	10.48	0.48
4	4	4	3.60	6.06	6.12	8.05	0.73	9.36	0.62
4	4	6	3.03	6.59	6.82	7.51	0.75	11.60	0.53
4	4	8	2.59	7.03	7.33	7.16	0.74	11.60	0.45

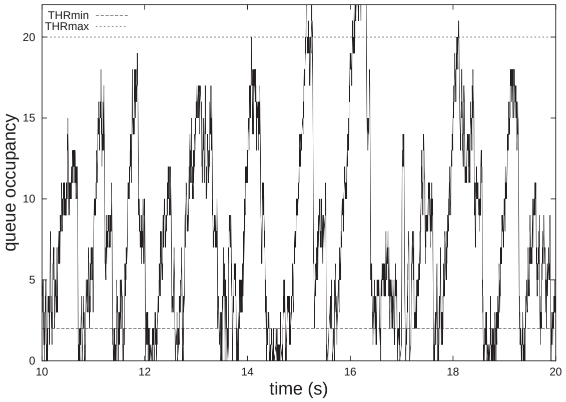

Fig. 8. Instantaneous node 4 buffer occupancy in a two bottleneck scenario

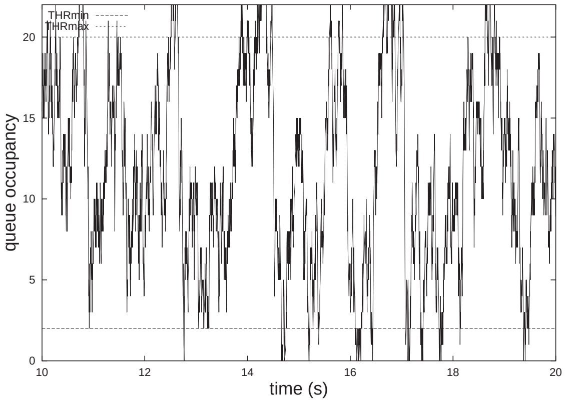

Fig. 9. Instantaneous node 2 buffer occupancy in a two bottleneck scenario

The normalized oscillation values in Table II confirm quantitatively the feelings obtained from Figures 8 and 9.

Note that, though the number of flows at each node and the flow round trip time should assure stable operation, instability arises due to the traffic aggregate from 6 to 7, which traverses both the congested links.

This example shows the limits of local AQM configuration ignoring the distributed nature of TCP flows control in a multiple bottleneck scenario. If we consider the configuration rules given in [9], instability probably does not arise in such a simple example, but there is a reduction of stability margins. This modifies the system dynamic response and reduces the system robustness to the flows number and the round trip time variation.

IV. THE ANALYTICAL INSIGHT

In this section, we provide an insight into the physical causes of instability in our counter-example. We start from a nonlinear multidimensional model of the network with some simplifying assumptions, and prove that the system is instable. Then, we come back to one-dimensional systems, by considering only one TCP aggregate at a time, the other ones acting as non reactive flows. Despite such system decoupling is not correct from an analytical point of view, it allows us to get again the linear system described in Section II, but with some different parameters. Hence, the effect of multi-bottleneck can be helpfully seen as a parameter variation in the same single bottleneck model we considered to configure the RED. It allows us to understand why instability arises and to simply predict the effect of some network scenario changes, such as the number of flows and the propagation delays. The limits of such an approximation are detailed in the following subsection.

A. Nonlinear Model

We extend the single bottleneck congestion model described in Section II to the case of two congested nodes. With reference to the network topology depicted in Figure 2 we obtain

$$\begin{cases} \dot{W}_5 = \frac{1}{R_5} - \frac{W_5 W_5(t-R_5)}{2R_5(t-R_5)} p_4(t-R_5) \\ \dot{W}_6 = \frac{1}{R_6} - \frac{W_6 W_6(t-R_6)}{2R_6(t-R_6)} (p_2(t-R_6) + \\ \qquad + p_4(t-R_6) - p_2(t-R_6) p_4(t-R_6)) \\ \dot{W}_8 = \frac{1}{R_8} - \frac{W_8 W_8(t-R_8)}{2R_8(t-R_8)} p_2(t-R_8) \\ \dot{q}_4 = \frac{W_5}{R_5} N_5 + \frac{W_6}{R_6} N_6 - 1_{q_4} C_4 \\ \dot{q}_2 = \frac{W_6}{R_6} N_6 + \frac{W_8}{R_8} N_8 - 1_{q_2} C_2 \end{cases} \tag{6}$$

where $R_5 = T_{p2} + \frac{q_4}{C_4}$, $R_8 = T_{p1} + \frac{q_2}{C_2}$, $R_6 = T_{p1} + \frac{q_2}{C_2} + \frac{q_4}{C_4}$. For sake of simplicity in (6), the time dependance is indicated only for delayed function values.

The above model relies essentially on the assumptions of the original single bottleneck model. One further limit is the way node 6 traffic has been considered in queue 4 equation: this equation ignores i) the delay from queue 2 to queue 4, and ii) that this traffic comes from another congested node, and therefore has been shaped by queue 2 (the outgoing traffic cannot overcome the link capacity between nodes 2 and 3).

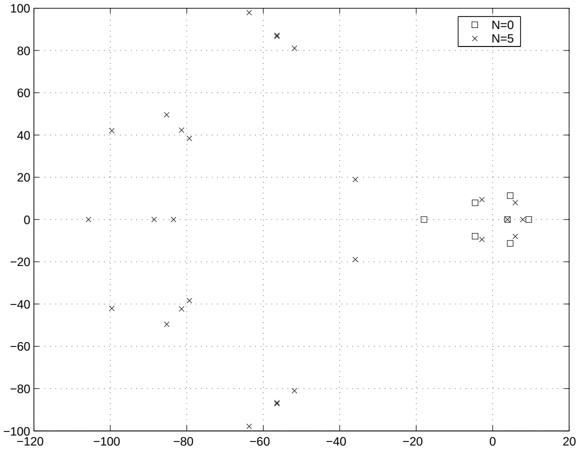

Fig. 10. Poles of the multidimensional system with two Padé approximations of time delays

By linearizing the Model 6 and using Padé functions to approximate time-delays, we obtain a rational Linear Time Invariant model. Thus, we study system stability considering the poles. Figure 10 shows the system poles for two different Padé approximations. The zero-order approximation ($N = 0$) simply corresponds to neglecting time-delays ($e^{-sR} \simeq 1$). There are four poles with positive real part, hence the system is unstable. As the approximation order increases, the number of poles increases, and, at least up to the 20th order approximation, there are always four poles with positive real part. For example poles for the fifth order approximation are shown in Figure 10.

B. One-dimensional models

Now, we consider individually each of the three aggregates and assume the other flows are non reactive ones, i.e., we focus on W_i, and assume $W_j/R_j = W_{j0}/R_{j0} =$ cost, for $j \neq i$, where $N_j W_{j0}/R_{j0}$ is the average throughput of the aggregate j. Due to congestion at nodes 2 and 4, $N_5 W_{50}/R_{50} + N_6 W_{60}/R_{60} \simeq C_4 = C_2 \simeq N_8 W_{80}/R_{80} + N_6 W_{60}/R_{60}$. We can derive the following model for the three aggregates:

$$\begin{cases} \dot{W}_5 = \frac{1}{R_5} - \frac{W_5 W_5(t-R_5)}{2R_5(t-R_5)} p_4(t-R_5) \\ \dot{q}_4 = \frac{W_5}{R_5} N_5 + \frac{W_{60}}{R_{60}} N_6 - 1_{q_4} C_4 \end{cases} \tag{7}$$

$$\begin{cases} \dot{W}_6 = \frac{1}{R_6} - \frac{W_6 W_6(t-R_6)}{2R_6(t-R_6)} (p_2(t-R_6) - \\ \qquad + p_4(t-R_6) - p_2(t-R_6) p_4(t-R_6)) \\ \dot{q}_4 = \frac{W_{50}}{R_{50}} N_5 + \frac{W_6}{R_6} N_6 - 1_{q_4} C_4 \\ \dot{q}_2 = \frac{W_6}{R_6} N_6 + \frac{W_{80}}{R_{80}} N_8 - 1_{q_2} C_2 \end{cases} \tag{8}$$

$$\begin{cases} \dot{W}_8 = \frac{1}{R_8} - \frac{W_8 W_8(t-R_8)}{2R_8(t-R_8)} p_2(t-R_8) \\ \dot{q}_2 = \frac{W_8}{R_8} N_8 + \frac{W_{60}}{R_{60}} N_6 - 1_{q_2} C_2 \end{cases} \tag{9}$$

The first and the third system equations are the same of the previous single-bottleneck system: bottleneck capacities are respectively equal to $C_{5eq} = C_4 - N_6 W_{60}/R_{60} = N_5 W_{50}/R_{50}$ and $C_{8eq} = C_2 - N_6 W_{60}/R_{60} = N_8 W_{80}/R_{80}$. Neglecting the product $p_2 p_4$ in comparison to the terms p_2 and p_4, the second system can be brought back to the single-bottleneck model too (as regards dynamics), where the bottleneck capacity is $C_{6eq} = C_4 - N_5 W_{50}/R_{50} = C_2 - N_8 W_{80}/R_{80} = N_6 W_{60}/R_{60}$ and the RED dropping curve is the sum of the two RED dropping curves at node 2 and 4, i.e. $p_{eq}(x) = p_2(x) + p_4(x)$.

The previous results are quite intuitive. Nevertheless, we can obtain them via linearization of Systems 7, 8 and 9. Calculations are similar to those detailed in the Appendix I of [9]. Thus, we obtain the following open-loop transfer functions:

$$F_5(s) = \frac{L \frac{(R_{50} C_{5eq})^3}{(2N_5)^2} e^{-sR_{50}}}{\left(1 + \frac{s}{K}\right) \left(1 + \frac{s}{\frac{2N_5}{R_{50}^2 C_{5eq}}}\right) \left(1 + \frac{s}{\frac{1}{R_{50}}}\right)} \tag{10}$$

$$F_6(s) = \frac{L_{eq} \frac{(R_{60} C_{6eq})^3}{(2N_6)^2} e^{-sR_{60}}}{\left(1 + \frac{s}{K}\right) \left(1 + \frac{s}{\frac{2N_6}{R_{60}^2 C_{6eq}}}\right) \left(1 + \frac{s}{\frac{1}{R_{60}}}\right)} \tag{11}$$

$$F_8(s) = \frac{L \frac{(R_{80} C_{8eq})^3}{(2N_8)^2} e^{-sR_{80}}}{\left(1 + \frac{s}{K}\right) \left(1 + \frac{s}{\frac{2N_8}{R_{80}^2 C_{8eq}}}\right) \left(1 + \frac{s}{\frac{1}{R_{80}}}\right)} \tag{12}$$

where $L_{eq} = 2L$. These transfer functions differ from transfer function in (5), only for the parameter values.

C. Stability considerations

In this section, we justify instability results shown in Section III, by applying the Nyquist criterion to the open-loop transfer functions in (10), (11) and (12).

We remember that our RED configuration assure stability for the system whose transfer loop function is (5) with $N = 8$, $R = 110$ms and $C = C_4 = C_2$.

From the new open-loop transfer functions, we see that the decrease of the number of effective flows for all the three aggregates and the increase of the RED slope for the aggregate 6 contribute to system instability. Yet, the decrease of the equivalent capacity makes the system more stable. In order to evaluate the dominating effect we have to consider numerical values for the parameters, but we can state that as the number of flows N_8 increases, W_5 exhibits instability. As the number of flows N_8 increase, the aggregate 6 is going to be harder choked, hence C_{5eq} approaches C_4 and the Nyquist plot corresponding to the transfer function (10) approaches the dashed curve in Figure 3, which corresponds to $N = 4$; the plot encircles the point $(-1, 0)$ and the corresponding closed loop system is unstable.

With the numerical values from Table II, the one-dimensional models predict that W_5 is unstable, whereas W_6 and W_8 are stable: W_8 is stable due to smaller RTT in comparison to aggregate 5 ($T_{p1} \leq T_{p2}$); as regards the window size W_6 a smaller C_{6eq} compensates the N reduction and L increase.

As regards the instability of the multidimensional system, all the variables show instability. As a matter of fact, W_5 instability implies the q_4 oscillations and hence the p_4 oscillations. The last affect the throughput of the aggregate 6. Aggregate 6 couples the two queues and hence it yields instability to q_2, and so on.

One-dimensional models allows us to simply predict for example the effect of increasing N_8. We have already stated that W_5 instability increases, at the same time W_8 becomes more stable and the coupling between the two queues by the aggregate 6 reduces. Hence, we expect an overall more stable behavior at queue 2. Performance metrics in Table II for $N_8 = 6$ and $N_8 = 8$ confirm results of one-dimensional models: instability increases at the downstream node and it decreases at the upstream one.

As regards the validity of our simple analysis, let us consider for example W_5. Results from System 7 are more accurate as long as i) aggregate 6 is small ($W_6(t) << W_5(t)$), or ii) it is not small, but it is not markedly affected by the dynamics of the aggregate 5 and of the queue 4, i.e. as long as the behavior of the aggregate 6 is determined elsewhere, in our example at the congested node 2. For example the model provides better a approximation if the number of flows N_8 increases or the round trip time R_8 decreases.

V. Guidelines for new AQM schemes

In the previous section we have modeled the effect of a multi-bottleneck path as a parameter variation for the single bottleneck model. This suggests that the effect of multi-bottleneck path can be counteracted by robust configuration of AQM controllers. In particular the minimum number of flows N^- should not take into account flows being controlled by other nodes.

Hence the network administrator should evaluate not only the minimum number of flows at each node and their round trip time, but he also should get more sophisticated information about traffic matrix across the network and contemporaneously congested nodes.

Another approach would be to implement new cooperative AQM controllers, that base their control action on information about the congestion status of the other nodes. Simplicity is an obvious requirement, particularly for signalling among nodes.

We think that the Explicit Congestion Notification (ECN) field in IP packets could be usefully employed for inter-nodes signalling.

According to [13], AQM controllers can set a Congestion Experienced (CE) codepoint in the packet header instead of dropping the packet, when such a field is provided in the IP header and understood by the transport protocol. The use of the CE codepoint allows the receiver(s) to receive the packet, avoiding the potential for excessive delays due to retransmissions after packet losses. Upon the receipt of a single packet with the CE codepoint set (CE packet), the ECN-capable TCP at the end-systems must react as it would do in response to a single dropped packet. Ultimately the ECN-capable TCP is required to halve its congestion window for any window of data containing either a packet drop or an ECN indication.

ECN has been proposed as a light in-band signalling form between nodes and client. We propose to employ it for inter-nodes signalling. It appears to be a simple way for nodes to transmit downstream information about their congestion status. The advantages of ECN employment are: no further network transmission resources are required, information travels along the data path, and it can be used by all the nodes controlling the flow.

AQM controller should monitor the ingoing traffic, and evaluate the share of traffic controlled elsewhere, by the percentage of CE packets.

As regards the way to employ such information, we can think to employ one of the well-known AQM schemes, where a tunable parameter can be set according to the controlled traffic share. For example a RED controller could decrease the dropping curve slope L as the percentage of CE packets increases in order to maintain a stable operation.

VI. Conclusions

In this paper we showed that RED configuration based on a single-bottleneck assumption may not prevent from traffic instability when congestion occurs, at the same time, in two different locations of the network. This motivates future works where we will design cooperative congestion local controllers under the assumption that a congested node may communicate its state to the neighbors.

References

[1] B. Braden, D. Clark, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, Recommendations on queue management and congestion avoidance in the internet, RFC 2309, April 1998.

[2] S. Floyd, V. Jacobson, ”Random Early Detection Gateways for Congestion Avoidance”, IEEE ACM Transactions on Networking, 1993

[3] S. Floyd, R. Gummadi, S. Shenker, ”Adaptive RED: An Algorithm for Increasing the Robustness of RED’s Active Queue Management”, August 1, 2001, under submission

[4] S. Athuraliya, V. H. Li, S. H. Low and Q. Yin, ”REM: Active Queue Management”, IEEE Network, May 2001

[5] W. Feng, D. Kandlur, D. Saha, K. Shin, ”Blue: A New Class of Active Queue Management Algorithms”, UM CSE-TR-387-99, 1999

[6] M. Christiansen, K. Jeffay, D. Ott, F.D. Smith, Tuning RED for web traffic, in Proceedings of ACM/SIGCOMM, 2000.

[7] M. May, T. Bonald, J. C. Bolot, Analytic Evaluation of RED Performance, in Proceedings of IEEE/INFOCOM, 2000.

[8] V. Firoiu, M. Borden, A Study of Active Queue Management for Congestion Control, in Proceedings of IEEE/INFOCOM, 2000.

[9] C. V. Hollot, V. Misra, D. Towsley, W. Gong “A Control Theoretic Analysis of RED” *IEEE INFOCOM*, 2001.

[10] V. Misra, W. Gong, D. Towsley, Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED, in Proceedings of ACM/SIGCOMM, 2000.

[11] C. V. Hollot, V. Misra, D. Towsley, W.-B Gong “Analysis and Design of Controllers for AQM Routers supporting TCP Flows” *IEEE ACM Trans on Automatic Control*, pp., vol.47, no.6, 2002.

[12] Network Simulator http://www.isi.edu/nsnam/ns/

[13] The Addition of Explicit Congestion Notification (ECN) to IP. Ramakrishnan, K.K., Floyd, S., and Black, D. RFC 3168, Proposed Standard, September 2001